

### REMARKS

Claim 1 is pending in the application and stands rejected under 35 USC §103 as being obvious in view of various references listed below.

To support a rejection under §103, the Examiner carries the burden of establishing a prima facie case for obviousness. To do so three basic criteria must be met.

First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest all the claim limitations. Consequently, the mere fact that a reference can be modified does not render the resultant modification obvious unless the prior art also suggests the desirability of the modification. MPEP § 2143. The Examiner carries the burden of providing *objective evidence* and making *specific factual findings* with respect to the motivation to modify a reference. MPEP § 2143.01.

### Claim Rejections

The Examiner rejected Claim 1 under 35 USC § 103 as being anticipated by USPN 6,594,697 issued to Praitis in view of USPN 6,523,062 issued to Bridgman. Claim 1 is directed to a method for generating error messages in a web based application and includes the following elements and limitations

- (a) searching said application for a predetermined error number;
- (b) retrieving an error message corresponding to said error number;
- (c) applying said error message to a style sheet in an error form; and
- (d) displaying said error form on a requesting device.

Rejecting Claim 1, the Examiner admits that Praitis does "not expressly teach applying said error message to a style sheet in an error form." For this, the Examiner relies on Bridgman stating that "Bridgman disclosed a method of applying the error message to a style sheet in an output document [error form] (Fig. 2, column 2 lines 29-65)." Furthermore, the Examiner, citing Praitis, col. 1, lines 27-37 and col. 6, lines 35-47, asserts that Praitis

Attorney Docket Number  
EXTS128

suggests "exploration of art and/or provided a reason to modify the method with the style sheet feature."

Contrary to the Examiner's assertions, Bridgman does not teach or suggest applying an error message to a style sheet in an error form in the manner required by Claim 1, and Praitis does not suggest or provide any motivation to modify its teachings. First, the section of Bridgman relied upon by the Examiner is reproduced as follows.

Authors creating documents as WML decks typically create the decks to be small in size, to accommodate the memory and processing limitations inherent in the target client devices. However, problems often arise when a user of a WAP-enabled device requests a document that was not created specifically with the wireless device limitations in mind. For example, it is becoming commonplace for XML documents (created irrespective of the client device) to be transcoded or otherwise transformed for downloading. When the target device is a relatively powerful machine with a large storage capacity such as a desktop computer, then downloading the XML document is unlikely to create problems. However, when the target device is a constrained device, then there may not be sufficient space for receiving and storing the document on the device. In addition, the processing capabilities of a constrained device may be insufficient for a document created without regard to the limitations of these devices.

Extensible Stylesheet Language ("XSL") style sheets provide an efficient means of filtering documents (such as XML documents), by defining translations on an input document that create only a specific set of desired document elements in the resulting output document. As is known in the art, a "style sheet" is a specification of a style that is to be used when presenting a document. Style sheets may also be utilized to describe transformations from one document type to another, such as transforming an XML document to a WML document. Style sheets may also be used as filters which describe transformations to reduce the amount of document content while maintaining the original document type. However, XSL style sheet filtering does not provide a means for limiting the size of the output document. WML decks in excess of 1 kilobyte in size may overload a constrained storage device, leading to undefined behavior when the device attempts to process the deck. From a user's perspective, it is unacceptable to allow this undefined behavior to occur.

Bridgman, col. 2, lines 29-65. Nothing in this section teaches applying an error message to a style sheet in an error form in the manner required by Claim 1.

*Attorney Docket Number  
EXTS128*

Next, the sections relied upon by the Examiner to establish motivation for combining Praitis and Bridgman or otherwise modifying Praitis are reproduced as follows.

The worldwide web, i.e., web, is a specific Internet network using a specific Internet protocol, i.e., Hyper Text Transfer Protocol (HTTP). In general, protocols are a set of rules in a prearranged data format defining how two computers communicate on the computer network 54. Servers that use the web are known as web servers and typically provide many separate electronic files, displays or documents that are accessible to other web servers or web clients. These electronic files are identified by a uniform resource identifier (URI) or a uniform resource locator (URL).

Praitis, col. 1, lines 27-37.

The browser displays the electronic document 60 that the user is currently viewing in the document display area 72. If the electronic document is too large to completely fit within the document area 72 the browser displays a portion of the document referred to hereafter as the "visible portion" in the document area 72 and presents the scroll bar 74 in the browser frame 70. The user can manipulate the scroll bar 74 with a mouse or other pointing device or input key commands on the keyboard to change the visible portion of the document that is shown by the browser within the document display area 72. The display 68 also comprises an address bar 75. The address bar displays the URL for the document 60 currently being displayed in document area 72.

Praitis, col. 6, lines 35-49. Nothing in these sections suggests any motivation to modify the teachings of Praitis "with the style sheet feature" in the manner required by Claim 1.

The cited sections do not supply even the smallest hint as to the use of style sheets.

For at least these reasons, Claim 1 is felt to distinguish over Praitis and Bridgman.

The Examiner also rejected Claim 1 under 35 USC § 103 as being anticipated by USPN 6,526,529 issued to Miksovsky in view of USPN 6,585,778 issued to Hind. Rejecting Claim 1, the Examiner admits that Miksovsky does "not expressly teach applying said error message to a style sheet in an error form." For this, the Examiner relies on Hind stating that "Hind disclosed a method of applying the error message to a style sheet in an output document [error form] (Abstract, Figures 2-7, 7, column 7 lines 19-50, column 7, line 65-column 8 line 57)." Furthermore, the Examiner, citing Miksovsky, col. 2, line 63-column 3, line 6, asserts that Miksovsky suggests "exploration of art and/or provided a reason to

modify the method with the style sheet feature to display output documents for used with hand-held devices."

Contrary to the Examiner's assertions, Hind does not teach or suggest applying an error message to a style sheet in an error form in the manner required by Claim 1, and Miksovsky does not suggest or provide any motivation to modify its teachings. First, the sections of Bridgman relied upon by the Examiner is reproduced as follows.

Enforcing data policy using style sheet processing. A Document Type Definition (DTD) associated with an Extensible Markup Language document is modified to specify a reference to stored data policy to be applied to document elements. Each data element may specify a different data policy. This technique uses minimal network transmission overhead, as the policy itself is not transmitted through the network until the DTD reaches the node where the data policy will be applied. Programming code implementing the data policy is then retrieved, using the references, by an Extensible Stylesheet Language (XSL) processor instrumented according to the present invention. Data policy is preferably enforced by overriding the existing XSL "value-of" method. DTD information describing a document element may be suppressed from a DTD being generated for the output document of the data policy enforcement process, providing privacy protection for the details of the associated policy.

Hind, Abstract.

The present invention defines a novel technique for enforcing data policy in a distributed network computing environment using style sheet processing. Preferably, this processing occurs at an intermediary in the delivery chain between a client who has requested stored data and the server application which has retrieved the requested information. Intermediaries commonly apply various types of translations and/or transformations based upon target context. For example, the Extensible Markup Language (XML) is widely adopted as an industry standard for the publishing and exchange of data through networks such as the Internet. When data is being transmitted in the form of an XML document, a common translation is to reformat the document into a different markup language, where the target markup language is better suited to the target context. Suppose the requesting user Sam from the previously-discussed example has requested data from his cell phone over a wireless connection. In this case, the target context comprises the user Sam; his limited-function, constrained device, the wireless network connection; and the browser or other application software from which Sam issued his request. It may be determined that Sam's browser does not support XML, but does support WBXML ("Wireless Application Protocol Binary XML"), which is a compact binary representation of XML developed for the purpose of document presentation for users of wireless computing devices. Thus, the intermediary would perform an XML to WBXML

*Attorney Docket Number*  
*EXTS128*

translation, and send the resulting WBXML document to the requesting user Sam. A typical means of performing this type of translation, as well as many other translations and transformation, is by applying a style sheet to the input document.

Hind, col. 7, lines 19-50.

A "style sheet" is a specification of a style that is to be used when presenting a document. The style specification includes information such as the font and margins to be used, the formatting layout, and other types of information that indicate how the presented document should appear. Style sheets may also be utilized to describe transformations from one document type to another (e.g. from XML to WML) or as filters which describe transformations to reduce the amount of content while maintaining the original document type.

One type of style sheet is an XSL Style Sheet. XSL Style Sheets are style sheets specified in XSL, which is a particular style sheet language. "XSL" is an acronym for "Extensible Stylesheet Language". An XSL Style Sheet specifies how an XML document is to be transformed, resulting in a different document which may or may not maintain the original document type. (Refer to "Extensible Stylesheet Language (XSL), W3C Working Draft 21 April 1999", referred to hereinafter as "the XSL Specification", and "XSL Transformations (XSLT), Version 1.0, W3C Working Draft 9 July 1999", which are available on the Web from the World Wide Web Consortium, or "W3C", for more information on using XSL for formatting and transforming documents)

Style sheets include "template rule" constructs, which define an input pattern and a template (also known as an "action") to use in creating an output result tree fragment. When applying a style sheet, the patterns in the template rules are matched against the syntax of the source document. When a match is found with the pattern, an output document fragment is created according to the actions specified in the template (which may include processing additional elements in the source document beyond the matching element). The source document is parsed recursively, until no more matching patterns are found. The resulting document fragments are then aggregated to yield a complete output document. (For more information on this process, refer to section 2, "Tree Construction", in the XSL Specification.) It is this template rule matching and substitution of different document elements according to the actions in the matching rules that enables style sheets to transform documents.

Style sheets may be written to search for and extract a specific subset of the information contained in an XML document. Or, a style sheet might tailor the information so that it can be delivered to a particular device, transforming the document for the characteristics of the device (such as which browser will be used to render the document, the screen size of the device, whether the screen supports color or grayscale, etc.). These techniques are well known in the art. (While the term "document" is

*Attorney Docket Number  
EXTS128*

used herein when discussing encoded data and application of style sheets thereto, it is to be understood that the information on which a style sheet operates may represent any type of information, and is not limited to the traditional interpretation of the word "document". As one example, a style sheet may be used to process an encoded representation of records from a data repository which specify a company's sales data. As another example, a style sheet may be used to format employee information retrieved from a corporate database for presentation. For ease of reference, the term "document" will be used herein to refer to these diverse types of information.)

Hind, col. 7, line 65 through col. 8, line 57. While these sections discuss style sheets, they do not teach or suggest applying an error message to a style sheet in an error form in the manner required by Claim 1.

Next, the section relied upon by the Examiner to establish motivation for combining Miksovsky and Hind or otherwise modifying Miksovsky is reproduced as follows.

Moreover, those skilled in the art will appreciate that the invention may be practiced with other computer system configurations, including hand-held devices, multi-processor systems, microprocessor-based or programmable consumer electronics, network PCs, minicomputers, mainframe computers and the like. The invention may also be practiced in distributed computing environments where tasks are performed by remote processing devices that are linked through a communications network. In a distributed computing environment, program modules may be located in both local and remote memory storage devices.).

Miksovsky, col. 2, line 57 through col. 3, line 6. Nothing in these sections suggests any motivation to modify the teachings of Praitis "with the style sheet feature" in the manner required by Claim 1. The cited section does not supply even the smallest hint as to the use of style sheets.


For at least these reasons, Claim 1 is felt to distinguish over Miksovsky and Hind.

*Attorney Docket Number*  
*EXTS128*

Conclusion

In view of the foregoing amendments and remarks, Applicant respectfully submits that the Examiner has failed to establish a prima facie case for obviousness and that Claim 1 is allowable over the cited art. Consequently, early and favorable action allowing these claims and passing the application to issue is earnestly solicited. The foregoing is believed to be a complete response to the outstanding Office Action.

Respectfully submitted,



Jack H. McKinney  
Attorney for Applicants  
Registration No. 45,685  
(208) 433-1991